

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 2000年 3月22日

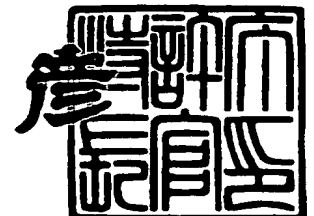
出 願 番 号
Application Number: 特願2000-084706

出 願 人
Applicant(s): 株式会社日立製作所

2000年 6月29日

特 許 庁 長 官
Commissioner,
Patent Office

近 藤 隆 彦



出証番号 出証特2000-3052512

【書類名】 特許願

【整理番号】 K99009331

【提出日】 平成12年 3月22日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 12/00

【請求項の数】 11

【発明者】

 【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

 【氏名】 荒井 正人

【発明者】

 【住所又は居所】 愛知県尾張旭市晴丘町池上 1 番地 株式会社日立製作所 情報機器事業部内

 【氏名】 梶浦 敏範

【特許出願人】

 【識別番号】 000005108

 【氏名又は名称】 株式会社日立製作所

【代理人】

 【識別番号】 100075096

 【弁理士】

 【氏名又は名称】 作田 康夫

【手数料の表示】

 【予納台帳番号】 013088

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

【ブルーフの要否】 要



【書類名】 明細書

【発明の名称】 アクセス制御システム

【特許請求の範囲】

【請求項 1】

ファイルへの入出力手段と、
前記ファイルを格納する記憶手段と、
前記ファイルへのアクセス制御ポリシーを記述したポリシーファイルと、
当該ファイルアクセス要求の正当性を前記に基づき判定するアクセス制御手段と、

前記入出力手段を用いたファイルアクセス要求の発行を監視し、発行されたファイルアクセス要求を前記アクセス制御手段に伝え、正当性判定結果を前記アクセス制御手段から受信する監視手段と、を備え、

前記アクセス制御ポリシーとして、アクセス対象となるファイルについて、アクセス要求発行元とアクセス実行手段とアクセスタイプとを特定する情報を前記ポリシーファイルに記し、

前記監視手段は、アクセス要求発行元とアクセス実行手段とアクセスタイプとを特定する情報を用いて、前記発行されたファイルアクセス要求を伝えることを特徴とするアクセス制御システム。

【請求項 2】

請求項 1 記載のアクセス制御システムであって、

前記アクセス制御ポリシーは、前記ファイルに対して禁止されたアクセスタイプと、該禁止されたアクセスが発生した場合にアクセス要求発行元に返すべきエラーコードと、例外として該アクセス要求を許可されたアクセス実行手段とアクセス要求発行元とを特定する情報とからなることを特徴とするアクセス制御システム。

【請求項 3】

請求項 2 記載のアクセス制御システムであって、

前記アクセス制御ポリシーに記すアクセス実行手段は、プログラムであり、該プログラムのパス名と、該プログラムの特徴値との組合せで特定することを特徴

とするアクセス制御システム。

【請求項 4】

請求項 3 記載のアクセス制御システムにおいて、

さらに、ファイルアクセス要求内容を登録するアクセスログファイルを備え、
前記アクセス制御手段は、前記ファイルアクセス要求を前記ポリシーファイル
の記述と照合し、正当性の判定結果を前記監視手段に送信するとともに、

前記アクセス要求が許可されるものであれば、当該アクセス実行手段の特徴値
を、前記監視手段に送信し、

前記アクセス要求が前記アクセス制御ポリシーに違反するものであれば、該フ
ァイルアクセス要求内容を、前記アクセスログファイル登録することを特徴とす
るアクセス制御システム。

【請求項 5】

請求項 4 記載のアクセス制御システムにおいて、

さらにオープンファイルテーブルを備え、

前記アクセス制御手段により正当であると判定された前記ファイルアクセス要
求のアクセスタイプがオープンアクセスである場合、

前記アクセス制御手段からレスポンス情報として取得した、アクセスタイプと
、前記アクセス対象となるファイルと、前記アクセス要求発行元と、アクセス実
行手段とを特定する情報を、前記オープンファイルテーブルに登録する手段と、

前記アクセス要求としてリードアクセスまたはライトアクセスが発行された場
合は、

前記オープンファイルテーブルを検索し、前記アクセス要求の正当性を判定す
る手段を備える

ことを特徴とするアクセス制御システム。

【請求項 6】

請求項 5 記載のアクセス制御システムであって、

前記監視手段は、さらに、

前記オープンファイルテーブルに登録されていないリードアクセス要求または
ライトアクセス要求を検知した場合は、前記アクセス制御手段を介して当該アク

セス要求内容を前記アクセスログファイルに登録する手段を備えることを特徴とするアクセス制御システム。

【請求項 7】

請求項 6 記載のアクセス制御システムであって、
前記監視手段は、さらに、
ファイルクローズ要求を検知した場合には前記オープンファイルテーブルから該当する情報を削除する手段を備えることを特徴とするアクセス制御システム。

【請求項 8】

請求項 7 記載のアクセス制御システムであって、
前記監視手段は、さらに、
前記ファイルアクセス要求が正当であると判定された場合、前記前記監視手段は前記アクセス実行手段の特徴値を算出し、と前記アクセス制御手段から受信した前記特徴値とを比較する手段と、
一致した場合には前記アクセス要求を許可する手段と、
一致しない場合には前記ファイルアクセス要求を無効にすると共に、前記アクセス制御手段を介して当該ファイルアクセス内容を前記アクセスログファイルに登録する手段とを備えることを特徴とするアクセス制御システム。

【請求項 9】

ファイルへの入出力手段と排他的に使用する第 1 の記憶手段と管理する第 1 の OS と、排他的に使用する第 2 の記憶手段を管理する第 2 の OS と、前記第 1 の OS と第 2 の OS との間でデータ通信するための通信手段とを具備した情報処理システムにおいて、

前記第 1 の OS が管理する前記ファイル入出力手段へ発行されたファイルアクセス要求を監視する監視手段を前記第 1 の OS に設け、

前記ファイルアクセス要求の正当性をアクセス制御ポリシーに基づき判定するアクセス制御手段を前記第 2 の OS に設け、

前記監視手段は、前記アクセス要求を、前記通信手段を介して、前記アクセス

制御手段に伝え、正当性判定結果を、前記通信手段を介して、前記アクセス制御手段から受信することを特徴とする情報処理システム。

【請求項 1 0】

請求項 9 記載の情報処理システムにおいて、

前記第 2 の OS は、前記アクセス制御ポリシーとして、アクセス対象となるファイルについて、アクセス要求発行元とアクセス実行手段とアクセスタイプとを特定する情報を記すポリシーファイルを備えることを特徴とする情報処理システム。

【請求項 1 1】

ファイルへの入出力手段と、前記ファイルを格納する記憶手段とを備えた情報処理装置に読み込まれ、実行され、前記情報処理装置上にアクセス制御システムを構成させるプログラムと前記プログラムが使用するファイルとを格納した記憶媒体であって、

前記プログラムが使用するファイルは、アクセス制御ポリシーを記述したポリシーファイルであって、

前記アクセス制御ポリシーは、アクセス対象となるファイルについて、アクセス要求発行元とアクセス実行手段とアクセスタイプとを特定する情報を記したものであり、

前記プログラムは、アクセス制御プログラムと、監視プログラムとを備え、

前記アクセス制御プログラムは、前記情報処理装置に、

当該ファイルアクセス要求の正当性を前記アクセス制御ポリシーに基づき判定させ、

前記監視手段は、前記情報処理装置に、

前記入出力手段を用いたファイルアクセス要求の発行を監視させ、

発行されたファイルアクセス要求を、アクセス要求発行元とアクセス実行手段とアクセスタイプとを特定する情報を用いて、前記アクセス制御プログラムに伝えさせること

を特徴とする記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、情報処理装置が管理する情報を不正なアクセスから保護する場合に好適なアクセス制御技術に関する。

【0002】

【従来の技術】

一般のコンピュータシステムでは、マルチユーザー・マルチタスクOSが備えるユーザー認証機構と、該認証結果に基づいたファイルアクセス制御機構を用いて機密情報ファイルの保護を実現しているケースが多い。具体的には、前記OSが実装された情報処理装置を利用する際に、ユーザーは必ず自己のユーザーIDとパスワードを入力し、認証を受ける。前記情報処理装置が管理する全てのファイル各々には、ファイルリードやライト等のアクセスタイプ毎に、ユーザーIDとグループIDを用いてアクセス可能なユーザーを定義したアクセスコントロールリストがセキュリティ属性情報として割り当てられている。。ユーザーがアプリケーションプログラムを介してファイルへアクセスした場合、上記OSは、アクセス要求元となるユーザーのID及び該ユーザーが所属するグループIDを、アクセス対象となるファイルに割り当てられたアクセスコントロールリストと照合し、当該リストに前記ユーザーが含まれている場合に限りアクセスを許可するといった制御を行なう。

【0003】

一方で、インターネットを介した情報発信、情報収集や、各種サービス提供を行なう手段としてWorld Wide Web (WWW) のサービスが広く使われている。WWWでは、Hyper Text Transfer Protocol (HTTP) と呼ばれる通信プロトコルが、リクエストデータとレスポンスデータの転送に用いられている。また、WWWシステムではコンテンツの不正な差し替えや、ネットワークを経由した機密情報流出を未然に防ぐためのセキュリティ技術がいくつか用意されている。

【0004】

HTTPが有するセキュリティ機構として、基本認証（Basic Authentication）と呼ばれるものがある。基本認証では、認証情報としてユーザーIDとパスワードを予めWWWサーバに登録しておき、ブラウザを通じてユーザーから送信されるユーザーIDとパスワードを、前記認証情報と比較して認証を行なう。各コンテンツへのアクセス権限設定を記述したポリシーファイルと、該ポリシーに基づくアクセス制御機構も前記WWWサーバに実装されている。同様な機構を、Common Gateway Interface (CGI) プログラムに実装し、ユーザー認証とコンテンツへのアクセス制御を実現することも可能である。

【0005】

WWWでは、他のセキュリティ技術として、認証局から発行される証明書に基づき、ユーザーとWWWサーバの相互認証と通信データの暗号化を行なうことが可能である。この技術は、消費者のクレジットカード番号がネットワーク上に流れる一部のインターネットショッピングのようなサービスにおいて、必須の技術とされている。これらの技術によるユーザー（クライアント）認証は、WWWサーバ側のOSが具備するユーザー認証機構とは通常独立したものであるが、前記認証に用いるユーザーIDや、ユーザーの証明書を、OSが管理するユーザーアカウントに関連付けて使用することができるWWWサーバプログラムもある。つまり、認証されたユーザーは、OSのアクセス制御下でWWWコンテンツにアクセスする。これらのセキュリティ技術については、“Web Security: A Step-by-Step Reference Guide” Lincoln D. Stein著、Addison-Wesley Pub Co; (ISBN: 0201634899) などに記載されている。

【0006】

近年増加しているインターネットサービスプログラム等に潜むセキュリティホールやバグを利用した不正アクセスを解決する他の技術としては、常駐型のファイル監視プログラムを情報処理装置上に設け、定期的にファイルの破壊の有無をチェックする方法が、特開平10-069417号公報に開示されている。これと同様の技術として、特定のファイルについてそのサイズの増減やタイムスタン

ブを定期的にチェックするプログラムを情報処理装置上に設け、該ファイルの変化を検知するツールが、シェアウェアやフリーウェアとして幾つか公開されている。

【0007】

【発明が解決しようとする課題】

先に述べたセキュリティホールやバグに関する情報は、<http://www.cert.org/> 等で報告され、バグ修正用のプログラムも各メーカーから配布されるようになっている。しかし、攻撃者あるいは侵入者は、あらゆる手段を用いて外部ネットワークから情報処理装置への侵入を試みる。たとえば、前記マルチユーザー・マルチタスクなOSが具備するアクセス制御機構では、アクセス要求元のユーザーIDとその所属グループに基づいてアクセスの可否を判断しているため、OSの管理者のような強い権限をもつユーザーに成りすまして侵入すれば、システム中のいかなるファイルにもアクセスできるので、システム中の情報を書き換えたり、機密情報を盗み出すことができてしまう。

【0008】

また、前記ファイルの変化を定期的にチェックする手法では、ファイルが改ざんされた後の検出になるため、不正アクセスが発生したことは分かるが、それを未然に防ぐことができないと共に、ファイルの不正な読み出しについては事後検出もできない。

【0009】

また、前記WWWサーバやCGIプログラムを利用してコンテンツのアクセス制御機能を実現したシステムにおいては、各コンテンツへのアクセス権限設定を記述したポリシーファイルも、その他のファイルと同じく不正アクセスの対象になる。したがって、該ポリシーファイルが破壊あるいは改ざんされた場合には、前記アクセス制御機能が正常に働かない。

【0010】

本発明の目的は、アクセス要求元（サブジェクト）がいかなる権限をもつ場合でも、そのアクセス手段およびアクセス対象（オブジェクト）を制限することが可能なアクセス制御システム及びその方法を提供することにある。

【 0 0 1 1 】

本発明の他の目的は、不正なファイルアクセスを未然に防ぐことが可能なアクセス制御システム及びその方法を提供することにある。

【 0 0 1 2 】

本発明の他の目的は、アクセス権限設定を記述したポリシーファイルと、該ポリシーファイルに基づいてアクセス制御を施行するプログラムを、外部からの不正アクセスや攻撃から保護可能なアクセス制御システム及びその方法を提供することにある。

【 0 0 1 3 】

【課題を解決するための手段】

上記目的を達成するために、本発明では、アクセス制御ポリシーとして、より厳密にアクセス要求を特定する情報、すなわち、アクセス要求元とアクセス実行手段とアクセスタイプとを特定する情報を用いる。

【 0 0 1 4 】

さらに具体的には、特定のファイルへのアクセスを、特定のユーザーが特定のプログラムを用いた場合のみ許可するアクセス制御ポリシーを記述したポリシーファイルを作り、アクセス制御手段が、ファイルへのアクセスが発生したときに、前記ポリシーファイルの記述に従ってアクセスの正当性すなわち可否を判定する。前記アクセス制御ポリシーには、アクセス対象となるファイルの名称と、アクセスが許可されたユーザー名とプログラム名の組合せを、ファイルのオープン・リード・ライト・削除・リネームといったアクセスタイプ毎に予め規定しておく。

【 0 0 1 5 】

また、不正なファイルアクセスを未然に防ぐために、本発明では、前記ファイルアクセスを監視するファイル I / O (I n p u t / O u t p u t) フック手段を前記情報処理装置に設け、該ファイル I / O フック手段は、前記アクセスを検知した時に、当該アクセスタイプ及びアクセス対象となるファイルの名称と、該アクセスの要求元となるユーザー名及びプログラム名を取得して前記アクセス制御手段に送信する。前記アクセス制御手段は、受信内容を前記ポリシーファイル

の内容と照合し、該ポリシーに違反するアクセスであれば当該アクセスを無効にし、前記ファイル I / O フック手段を経由して前記アクセス要求元にエラーを返す。

【 0 0 1 6 】

また、前記ポリシーファイルと前記アクセス制御手段を保護するために、本発明では、1 台の情報処理装置上に 2 つの OS と、両 OS 間でデータ通信するためのプロセス間通信手段と、両 OS が互いに排他的に占有するメモリや磁気ディスクおよびネットワークデバイスを設ける。前記 2 つの OS のうち一方をアクセス監視対象となるサービス用 OS として使用し前記ファイル I / O フック手段を前記サービス用 OS のカーネルレベルのモジュールとして設ける。もう一方をセキュリティ用 OS として使用し、前記アクセス制御手段および前記ポリシーファイルを占有する磁気ディスクに格納すると共に、前記アクセス制御手段をプロセスとして動作させ、該アクセス制御手段とポリシーファイルを前記サービス用 OS 上の、サービスを受けるユーザが使うプロセスからアクセスできないようにする。

【 0 0 1 7 】

なお、本発明における OS とは、ユーザーまたはプログラムからの要求に応じて記憶媒体中のデータ、ファイルへのアクセスを実行する機能と、アクセス要求元のユーザーやプログラムを識別する機能と、排他的に占有する記憶手段を有するプログラムモジュールを意味しており、

- ・ データ（ファイル）アクセスを管理しており、検知が可能である。

【 0 0 1 8 】

- ・ アクセス要求元のユーザーを識別できる。

【 0 0 1 9 】

- ・ アクセス要求元のアプリケーションを識別できる。

【 0 0 2 0 】

といった特徴を持つ。したがって、一般的に OS と呼ばれるものに限らず上記特徴を持つものであれば、本発明を適用することは可能である。

【 0 0 2 1 】

【 発 明 の 実 施 の 形 態 】

以下、図を用いて本発明の実施の一形態を説明する。

図 1 は、本発明のアクセス制御システムの一構成例である。100 はサーバ情報処理装置であり、サービス用オペレーティングシステム（OS）103 が管理するメモリ 101、当該サービス用 OS 103 が占有するディスクコントローラ 112 a と磁気ディスク 114 a と LAN コントローラ 113、セキュリティ用 OS 104 が管理するメモリ 102 と、当該セキュリティ用 OS 104 が占有するディスクコントローラ 112 b と磁気ディスク 114 b その他を備える。

【 0 0 2 2 】

サーバ情報処理装置 100 の起動時には、サービス用 OS 103 とサーバプログラム 109 がディスクコントローラ 112 a を介して磁気ディスク 114 a からメモリ 101 上にロードされると共に、セキュリティ用 OS 104 とアクセス制御プログラム 110 がディスクコントローラ 112 b を介して磁気ディスク 114 b からメモリ 102 上にロードされる。前記サービス用 OS 103 中のプロセス間通信プログラム 108 は、ファイル I/O フックプログラムと前記アクセス制御プログラム 110 との間でデータ交換を行なうために使用するものとし、前記サーバプログラム 109 からは直接使用できないインタフェースとなっている。

【 0 0 2 3 】

また、サービス用 OS 103 は、流通している既存のマルチユーザー・マルチタスクな OS と同様、ユーザーの識別・認証機能を具備しているものとする。前記サーバ情報処理装置 100 と複数のクライアント情報処理装置 120 は、ローカルエリアネットワーク（LAN）115 を介して相互に接続されており、前記サーバプログラム 109 は、LAN コントローラ 113 と LAN 115 を介して、クライアントプログラム 121 からのリクエストの受信と、当該クライアントプログラム 121 へのレスポンスの送信を行なう。サーバプログラム 109 とクライアントプログラム 121 は、例えば WWW サーバとブラウザに相当するプログラムである。なお、前記サーバ情報処理装置 100 とクライアント情報処理装

置120は、電話回線やインターネットを介して接続されていてもよい。また、サービス用OSとセキュリティOSとは、OS間通信のインタフェースを非公開にすれば別々の装置上にあっても良く、セキュリティ的にも問題はない。

【0024】

前記サーバプログラム109のようにOS上で動作するアプリケーションプログラムは、一般にユーザーレベルのプロセスと呼ばれている。該サーバプログラム109は、前述のようにクライアントプログラム121からのリクエストに基づいて処理を実行するものであり、言い換えればネットワークからの攻撃の対象にもなり得るプログラムである。これに対し、ファイルI/Oフックプログラム106やファイルシステムドライバ107のようにOSの一機能として動作するプログラムは、一般にカーネルレベルのモジュールと呼ばれており、多くのコンピュータシステムではアクセス制御機能をカーネルレベルのモジュールとして実装している。

【0025】

図1のサーバ情報処理装置100のように、1台の情報処理装置上に複数のOSを同時に動作させるための技術としては、仮想計算機あるいはマイクロカーネルがある。その他、リアルタイムOSを一般的なOSのカーネルレベルのモジュールとして実装すると共に、前記リアルタイムOSがシステム障害を検知すると、リアルタイム処理を継続しながらもシステムを自動的に再起動する方法が、特開平11-024943号公報に開示されている。また、プロセス間通信プログラム108のような、異種OS上のプロセス間通信を実現する方法は、特開平11-085546号公報に開示されている。これらを本発明の前提条件として、引続き実施の形態を述べる。

【0026】

図2は、前記ポリシーファイル200のデータ構造を示したものである。

【0027】

210から212は、ポリシー記述の一例を示したものである。オブジェクト名201は、アクセスを制限すべきファイルの名称を示す。オブジェクト名として、210や212のようにファイル単位の指定と、211のようなディレクトリ

単位の指定が可能となっている。ディレクトリ単位の指定では、該ディレクトリ名に続くファイル名を所定の文字、記号（たとえばアスタリスク（*））で表記する。前記オブジェクトに対して禁止すべきアクセスを、禁止されたアクセスのタイプ202に示す。エラーコード203は、前記禁止されたアクセスのタイプ202が発生した際に、当該アクセス発行元（サブジェクト）となるプログラムに返すべきエラーコードを示す。例外サブジェクト204は、特別にアクセスを許されたプログラムの名称である。プログラムのハッシュ値205は、前記例外サブジェクトとして指定されたプログラムファイルの特徴値（ハッシュ値）をたとえば8バイトで表したものである。ユーザー名206は、前記例外サブジェクト204のプログラムを利用可能なユーザーを、前記サービス用OSが管理するユーザー名またはグループ名で表したものである。

【0028】

つまり前記ポリシーは、前記プログラムのハッシュ値205に登録されたハッシュ値を有するプログラムが前記例外サブジェクトとして登録され、且つ該プログラムをユーザー名206に登録されたユーザーあるいはグループのメンバーが利用している場合に限り、例外としてアクセスを許可することを表している。同時に、これら条件が整っていない場合は、アクセス発行元のプログラムに対して前記エラーコード203を返すことを意味する。

【0029】

前記ポリシーファイル200の設定は、システムのセキュリティ管理者が行なうものとする。例えば、HTMLファイルなどのWWWコンテンツを侵入者によって不正に書き換えられないようにするには、該HTMLファイルへのライトアクセスを原則として禁止しておき、例外としてコンテンツ管理者に任命されているユーザーが特定のHTML編集用プログラムを用いた場合に限りライトアクセスを許可するといったポリシーを前記ポリシーファイル200に記述すればよい。

【0030】

例外サブジェクトには、サーバプログラム109のように、ネットワークからの攻撃を受けやすいプログラムを指定しないことが重要である。更に例外サブジ

ェクトとして登録するプログラムを、CD-ROM等の取り外し可能な記憶媒体に格納しておき、必要なときだけ媒体を装着して使用すればより確実に保護できるので、さらなるセキュリティ効果が期待できる。図1の構成に当該媒体の駆動装置を追加し、ディスクコントローラ112aが当該駆動装置に対応すれば、このような記憶媒体が使用可能となる。

【0031】

図3は、アクセスログファイル300のデータ構造の一例を示したものである。アクセスログファイル300は、前記ポリシーファイル200にて禁じられているアクセスが発生した事実を、アクセス制御プログラム110のアクセスログ登録ルーチン406が書き込むためのファイルであり、当該アクセスが発生した日時301と、当該アクセスの対象となったファイルを表すオブジェクト名302と、当該アクセスのタイプ303、当該アクセスを発行したプログラムを表すサブジェクト名304、そして前記プログラムを利用していたユーザーを表すユーザー名305から構成される。

【0032】

図4に、ファイルI/Oフックプログラム106と、アクセス制御プログラム110の構成を示す。

ファイルI/Oフックプログラム106は、前記サーバ情報処理装置100の起動時に、サービス用OS103と共にメモリ101上にロードされる。アクセス制御プログラム110は、前記サーバ情報処理装置100の起動時に、セキュリティ用OS104と共にメモリ102上にロードされる。

【0033】

図5と図6を用いて、本発明のアクセス制御の概要を記す。

【0034】

図5は、I/Oマネージャ105と、ファイルI/Oフックプログラム106と、アクセス制御プログラム110の三者間における処理の流れを示すものである。501は、サーバプログラム109が発行したファイルアクセスが、I/Oマネージャ105を経由してファイルI/Oフックプログラム106に到達するまでの通信経路であり、サービス用OS103の上で動作する全てのアプリケー

ションプログラムからのファイルアクセスについて共通なものである。ファイル I/O フックプログラム 106 には、図 6 に示す I/O パケット 600 のデータが渡される。I/O パケット 600 は、ファイル I/O フックプログラム 106 に渡された時点では、既にサービス用 OS が具備するアクセス制御機構はパスしている。つまり、サービス用 OS が具備するアクセス制御によって許可されたアクセスに関するパケットである。

【0035】

図 6 は、前記三者間を流れるパケットのデータ構造を示したものである。601 はアクセス対象のファイルを表すオブジェクト名である。602 は前記オブジェクトへのアクセスタイプを示す。603 は、当該アクセスを発行したプログラムのプロセス ID を示す。604 は当該アクセスの処理結果を示すステータスを示す。605 は当該アクセスに関連するデータの長さを示すものであり、例えば前記アクセスタイプ 602 がファイルへの書込み（ライト）要求であれば、書込みデータがデータ領域 607 に格納され、当該書込みデータの長さがデータ長 605 に格納される。606 は、前記データ領域 607 へのポインタである。

【0036】

図 5 において、502 は、ファイル I/O フックプログラム 106 が検知したファイルオープン要求に関するアクセス権限チェックと、アクセスログ 300 への書き込みを、アクセス制御プログラム 110 に対してリクエストするに用いる通信経路である。前記リクエストのデータ構造を、図 6 のリクエストパケット 610 に示す。サブジェクト名 611 は、ファイルアクセスの発行元となるプログラムの名称であり、ユーザー名 612 は、前記プログラムの利用者をユーザー名とグループ名で表すものである。サブジェクト名 611 及びユーザー名 612 は、受信した I/O パケット 600 に含まれるプロセス ID 603 を指定して、前記サービス用 OS として用いる標準の OS に対してシステムコールを発行することで取得できる。

【0037】

前記リクエストパケット 610 は、プロセス間通信プログラム 108 が当該パケットのデータを、ファイル I/O フックプログラムのメモリ空間 630 からア

クセス制御プログラムのメモリ空間 6 4 0 に複写することで伝達される。アクセス制御プログラム 1 1 0 では、受信したリクエストパケットに含まれる情報を、ポリシーファイル 2 0 0 の内容と照合し、その結果をレスポンスパケット 6 2 0 に登録して前記ファイル I / O フックプログラムに伝達する。この伝達は、前記通信経路 5 0 2 と同様に、プロセス間通信プログラム 1 0 8 が前記パケットのデータを、アクセス制御プログラムのメモリ空間 6 4 0 からファイル I / O フックプログラムのメモリ空間 6 3 0 に複写することで達成される。5 0 3 は、当該レスポンスパケットの通信経路を示すものである。

【 0 0 3 8 】

ファイル I / O フックプログラム 1 0 6 は、受信したレスポンスパケット 6 2 0 の内容から当該アクセスの可否を判断し、アクセス不可であればエラーコード 2 0 3 を I / O パケット 6 0 0 のステータス 6 0 4 に設定し、当該 I / O パケットを前記 I / O マネージャ 1 0 5 に返す。前記ステータス 6 0 4 にエラーコードが設定されている場合、前記 I / O マネージャ 1 0 5 は、通信経路 5 0 6 を用いてファイルアクセス発行元のサーバプログラム 1 0 9 に当該エラーを返す。一方、前記ステータス 6 0 4 にエラーコードが設定されていなければ、通信経路 5 0 5 を用いてファイルシステムドライバ 1 0 7 及びディスクコントローラ 1 1 2 a を介して磁気ディスク 1 1 4 a へのファイルアクセスを続行し、通信経路 5 0 6 を用いて当該アクセスの結果を前記サーバプログラム 1 0 9 に返す。

【 0 0 3 9 】

本発明のアクセス制御の具体的処理内容を、図 7 から図 1 6 を用いて説明する。

図 7 は、ファイル I / O フックプログラム 1 0 6 が具備するファイル I / O フックルーチンの処理内容を示すフロー図である。

サーバプログラム 1 0 9 が、ステップ 7 0 1 でクライアントプログラム 1 2 1 からのリクエストを受信し、ステップ 7 0 2 で前記リクエストに応じたファイルアクセスをサービス用 OS 1 0 3 に対して発行した場合を想定する。

【 0 0 4 0 】

当該ファイルアクセスは、通信経路 5 0 1 を経由して、ファイル I / O フック

ルーチン400にI/Oパケット600として渡される（ステップ703）。ステップ703からステップ712は、ファイルI/Oフックルーチン400の処理フローである。ステップ704では、I/Oパケット600に含まれるプロセスID603から、ファイルアクセス発行元のサブジェクト名611とユーザー名612を取得する。

【0041】

ステップ705からステップ708では、アクセスタイプ602を調べ、ファイルオープン、ファイルクローズ、ファイルリード又はライト、またはファイル削除又はリネームにそれぞれ対応する処理ルーチンを実行する。これらに該当しない場合、又は対応する処理ルーチンを実行した後は、通信経路504を経由して、ステップ713にてI/Oマネージャ105の処理に戻る。

【0042】

図8を用いてオープン処理ルーチン401の処理フローを説明する。ステップ801では、アクセス日時として現在の日付と時刻を取得する。ステップ802では、リクエストパケット610を作成すると共に、当該パケット中のエラーコード203とハッシュ値205を0に初期設定する。次にステップ803にて、アクセス制御プログラム110のオープン制御ルーチン405をコールすると共に、前記リクエストパケット610をオープン制御ルーチン405に渡す。

【0043】

オープン制御ルーチン405の処理フローを、図9を用いて説明する。ステップ901では、受信したリクエストパケット610のうち、オブジェクト名601とアクセスタイプ602の内容を、ポリシーファイル200と照合する。ステップ902では、当該アクセスが禁止されたアクセスのタイプとして登録されているかどうかを判別する。このとき、禁止されたアクセスタイプに該当しなければ正当なアクセスとみなし、ステップ903にて、許可されたアクセスタイプをレスポンスパケット620の認可アクセス613に登録し、オープン処理ルーチン401の処理に戻る。ここで、許可されたアクセスタイプとは、オブジェクト名601で示されるファイルに対するリードとライトの内、サブジェクト名611で示されるプログラムから実行可能なアクセスタイプのことを表す。したがっ

て、オブジェクト名601で示されるファイルに対して、ポリシーファイル200の中でリードとライト共に許可されていれば、認可アクセス613にはリードとライトの両方を登録する。また、リードのみが許可されている場合はリードのみを、ライトのみが許可されている場合はライトのみを前記認可アクセス613に登録する。リードとライトの両方を禁じる場合には、ポリシーとして当該ファイルのオープンを禁止するよう、予めポリシーファイル200に記述しておく必要がある。

【0044】

ステップ902にて禁止されたアクセスだと判断した場合、ステップ904にて、前記サブジェクト名611とユーザー名612が、共にポリシーファイル200に記述された例外サブジェクト204とユーザー名206に該当するか否かを判別する。このとき、サブジェクト名については該プログラムファイルのパス名が完全に一致するか否かを判別する。またユーザー名については、ユーザー名612に含まれるユーザー名とグループ名のいずれかが一致するか否かを判別する。判別の結果、例外サブジェクト204かつユーザー名206に該当する場合、ステップ905にて当該プログラムファイルのハッシュ値205をポリシーファイル200から取得して、レスポンスパケット620に設定する。また、ステップ906では、ステップ903と同様に、許可されたアクセスタイプをレスポンスパケット620の認可アクセス613として設定する。その後、ステップ908で、該当するエラーコード203をレスポンスパケット620に設定してから、オープン処理ルーチン401へ戻る。ここでのエラーコード設定は、オープン処理ルーチン401のステップ804からステップ810の処理の中で意味を持つものである。

【0045】

ステップ904にて、例外サブジェクト204とユーザー名206とに該当しないと判断した場合、ステップ907にて、当該アクセスの内容をアクセスログファイル300に書込む。その後、ステップ908で、該当するエラーコード203をレスポンスパケット620に設定してから、オープン処理ルーチン401へ戻る。

【0046】

次に、図8において、ステップ804ではレスポンスパケット中のハッシュ値205の値を調べ、0であれば例外として認められたアクセスではないと判断し、ステップ805でエラーコードの値をチェックする。エラーコードが0でなければ禁止されたアクセスであったと判断し、ステップ806にて、I/Oパケット600のステータス604に、当該エラーコード203を設定して処理を終了する。ステップ805で、エラーコードが0であれば、正当なアクセスであったと判断し、ステップ810にて当該アクセス情報を後述するオープンファイルテーブルの先頭アドレスに登録してから処理を終了する。

【0047】

ステップ804でハッシュ値が0でない場合は、例外として認められたサブジェクトであると判断し、ステップ807にてサブジェクト名601で示されるプログラムファイルのハッシュ値を算出し、ステップ808にてレスポンスパケット中のハッシュ値205と比較する。両者が等しければ、例外サブジェクト204に相当するとみなして、ステップ812にて当該アクセス情報をオープンファイルテーブルの先頭アドレスに登録してから本ルーチンの処理を終了する。ハッシュ値が等しくなければ、前記プログラムファイルが不正なプログラムであるとみなして、ステップ809にて前記リクエストパケットにおけるサブジェクト名611を例外サブジェクトにならないように、例えばサブジェクト名としてnullデータに、変更し、ステップ810にてアクセスログ登録ルーチン406をコールすると共に前記リクエストパケット610をアクセスログ登録ルーチン406に渡す。

【0048】

該アクセスログ登録ルーチン406の処理フローを、図14を用いて説明する。ステップ1401で、リクエストパケット610の内容をポリシーファイル200と照合する。予め前記ステップ809で例外サブジェクト扱いにならぬようにサブジェクト名611をクリアしておくことにより、必ず禁止されたオープンアクセスとして扱われる。ステップ1402にて、該当するエラーコードをポリシーファイル200から読み出してレスポンスパケット620に設定し、当該パ

ケットを返す。ステップ1403にて当該アクセス内容を、アクセスログファイル300に書き込み、オープン処理ルーチン401の処理に戻る。

【0049】

図8のオープン処理ルーチン401において、ステップ811でレスポンスパケット中のエラーコード203をI/Oパケットのステータス604に設定して本ルーチンの処理を終了する。

【0050】

図10を用いてオープンファイルテーブルについて説明する。オープンファイルテーブル1000は、現在オープン中のファイルに関する情報を格納した構造体データ1002の集合である。1個の構造体には1件のオープンファイルの情報を記憶しており、ファイルI/Oフックプログラム106では、各構造体を先頭アドレス1001とポインタ1003によりリストとして管理する。本オープンファイルテーブル1000の一例を図11に示す。

【0051】

オープンファイルテーブル1000に登録された情報に該当するアクセスであれば、該アクセスを許可することになるため、オープンファイルテーブル1000を不正に書換えられないよう保護することが重要である。サービス用OS103として、プロセス毎に独立したメモリ空間が割当てられると共にメモリ空間の排他制御機構が働くOSを使うことで、前記オープンファイルテーブル1000も別プロセスからは不正に書換えできない仕組みにすることが可能になる。

【0052】

図12は、クローズ処理ルーチン402の処理フローを示したものである。ステップ1201では、受信したI/Oパケット600の中にあるオブジェクト名601と、プロセスID603、並びにプロセスID603から取得したサブジェクト名611およびユーザー名612の組み合わせと同じものをオープンファイルテーブル1000から検索する。ステップ1202で、該当する情報がテーブルにあれば、ステップ1203にて該当する情報をテーブルから削除する。一方、テーブルに存在しなければ、ポリシーファイル200に登録されていないファイルへのクローズ要求とみなし、本ルーチンの処理を終了する。

【0053】

図13は、リード・ライト処理ルーチン403の処理フローを表したものである。ステップ1301では、受信したI/Oパケット600の中にあるオブジェクト名601と、プロセスID603と、プロセスID603から取得したサブジェクト名611と、ユーザー名612との組み合わせと同じものをオープンファイルテーブル1000から検索し、更にアクセスタイプ602が認可されたアクセスタイプ613に含まれるか否かをチェックする。ステップ1302にて、前記オブジェクト名601と、プロセスID603と、サブジェクト名611と、ユーザー名612と、アクセスタイプ602との組合せがテーブル1000にあれば、本ルーチンの処理を終了する。このことは、オープンファイルテーブル1000に登録されたアクセスであれば正当なアクセスであるとみなし、アクセス制御プログラム110の処理を行わないことを意味する。これは、本発明のアクセス制御によって生じるシステムのパフォーマンス低下を軽減する効果がある。

【0054】

ステップ1302にて、オープンファイルテーブルに登録されたアクセスでなければ前記ポリシーファイル200により禁止されたアクセスとみなし、ステップ1303でアクセス日時を取得し、ステップ1304でリクエストパケット610を作成する。このとき、エラーコード203を0に初期設定する。この後、ステップ1305にてアクセスログ登録ルーチン406の処理にジャンプする。

【0055】

図14のアクセスログ登録ルーチン406において、ステップ1401ではリクエストパケット610の内容をポリシーファイル200と照合し、ステップ1402にて、該当するエラーコードをポリシーファイル200から読み出してレスポンスパケット620に設定する。次に、ステップ1403にて当該アクセス内容を、アクセスログファイル300に書き込み、リード・ライト処理ルーチン403の処理に戻る。リード・ライト処理ルーチン403では、図13のステップ1306において、レスポンスパケットで受信したエラーコードをI/Oパケットのステータス604に設定し、本ルーチンの処理を終了する。

【0056】

図15は、削除・リネーム処理ルーチン404の処理フローを表したものである。図8のオープン処理ルーチン401と同じ処理内容のステップには同じ番号を付している。

【0057】

ステップ1503にて、アクセス制御プログラム110の削除・リネーム制御ルーチン407をコールすると共に、前記リクエストパケット610を削除・リネーム制御ルーチンに渡す。

【0058】

削除・リネーム制御ルーチン407の処理フローを、図16を用いて説明する。図9のオープン制御ルーチン405と同じ処理内容のステップには同じ番号を付している。ステップ1602では、当該アクセスが禁止されたアクセスのタイプとして登録されているかどうかを判別する。禁止されたアクセスタイプに該当しなければ正当なアクセスとみなし、削除・リネーム処理ルーチン404の処理に戻る。

【0059】

ステップ1602にて禁止されたアクセスだと判断した場合、図9と同様にステップ904、905の処理を行い、ステップ1603にて、例外サブジェクト204とユーザー名206に該当しないと判断した場合、図9と同様にステップ907の処理を行う。

【0060】

ステップ905または907の処理を済ませると、ステップ908で、該当するエラーコード203をレスポンスパケット620に設定してから、削除・リネーム処理ルーチン404へ戻る。

【0061】

次に図15において、図8と同様にステップ804の処理を行う。ステップ1505では、エラーコードが0であれば、正当なアクセスであったと判断し、本ルーチンの処理を終了する。ステップ807にてサブジェクト名601で示されるプログラムファイルのハッシュ値を算出し、ステップ808にてレスポンスパ

ケット中のハッシュ値 2 0 5 と比較して、両者が等しければ、例外サブジェクト 2 0 4 に相当するとみなして本ルーチンの処理を終了する。ハッシュ値が等しくない場合は図 8 と同様の処理を行い、アクセスログ登録ルーチン 4 0 6 を呼び出す。当該ルーチンが処理を済ませると削除・リネーム処理ルーチン 4 0 4 の処理に戻る。

【 0 0 6 2 】

図 1 5 において、ステップ 1 5 1 1 でレスポンスケット中のエラーコード 2 0 3 を I / O パケットのステータス 6 0 4 に設定して本ルーチンの処理を終了する。

【 0 0 6 3 】

以上説明したように、本発明によれば、進入者が強い権限をもつユーザーに成りすました場合でも、ファイルへのアクセスを制限できるという効果がある。

また、不正なファイルアクセスを未然に防ぐことができるという効果がある。

また、前記ポリシーファイルとアクセス制御手段を、外部からの不正アクセスや攻撃から保護できるという効果がある。

また、図 1 に示すシステムの、サーバ情報処理装置とクライアント情報処理装置とを結ぶ LAN 1 1 5 などの通信回線上に、ファイアウォールを設け、さらに本発明を併用すれば、より強固なセキュリティを確保することが出来る。

【 0 0 6 4 】

【発明の効果】

本発明によれば、情報処理装置が管理するファイル、情報を不正なアクセスから保護することが可能になる。

【図面の簡単な説明】

【図 1】

本発明の実施の形態におけるアクセス制御システムの一構成例を示す図。

【図 2】

アクセス制御ポリシーの設定を格納するためのポリシーファイルを示す図。

【図 3】

不正アクセスが発生した事実を記録するためのアクセスログファイルを示す図

【図 4】

ファイル I / O フックプログラム 1 0 6 とアクセス制御プログラム 1 1 0 を構成するプログラムルーチンを示す図。

【図 5】

本発明の実施の一形態において、各モジュール間を流れるデータの通信経路を示す図。

【図 6】

本発明の実施の一形態において、各モジュール間を流れるデータの構造を示す図。

【図 7】

ファイル I / O フックルーチン 4 0 0 の処理のフローチャートを示す図。

【図 8】

オープン処理ルーチン 4 0 1 の処理のフローチャートを示す図。

【図 9】

オープン制御ルーチン 4 0 5 の処理のフローチャートを示す図。

【図 1 0】

オープンファイルテーブルの構造体とそのリストを示す図。

【図 1 1】

オープンファイルテーブルに登録されるデータの一例を示す図。

【図 1 2】

クローズ処理ルーチン 4 0 2 の処理のフローチャートを示す図。

【図 1 3】

リード・ライト処理ルーチン 4 0 3 の処理のフローチャートを示す図。

【図 1 4】

アクセスログ登録ルーチン 4 0 6 の処理のフローチャートを示す図。

【図 1 5】

削除・リネーム処理ルーチン 4 0 4 の処理のフローチャートを示す図。

【図16】

削除・リネーム処理ルーチン407の処理のフローチャートを示す図。

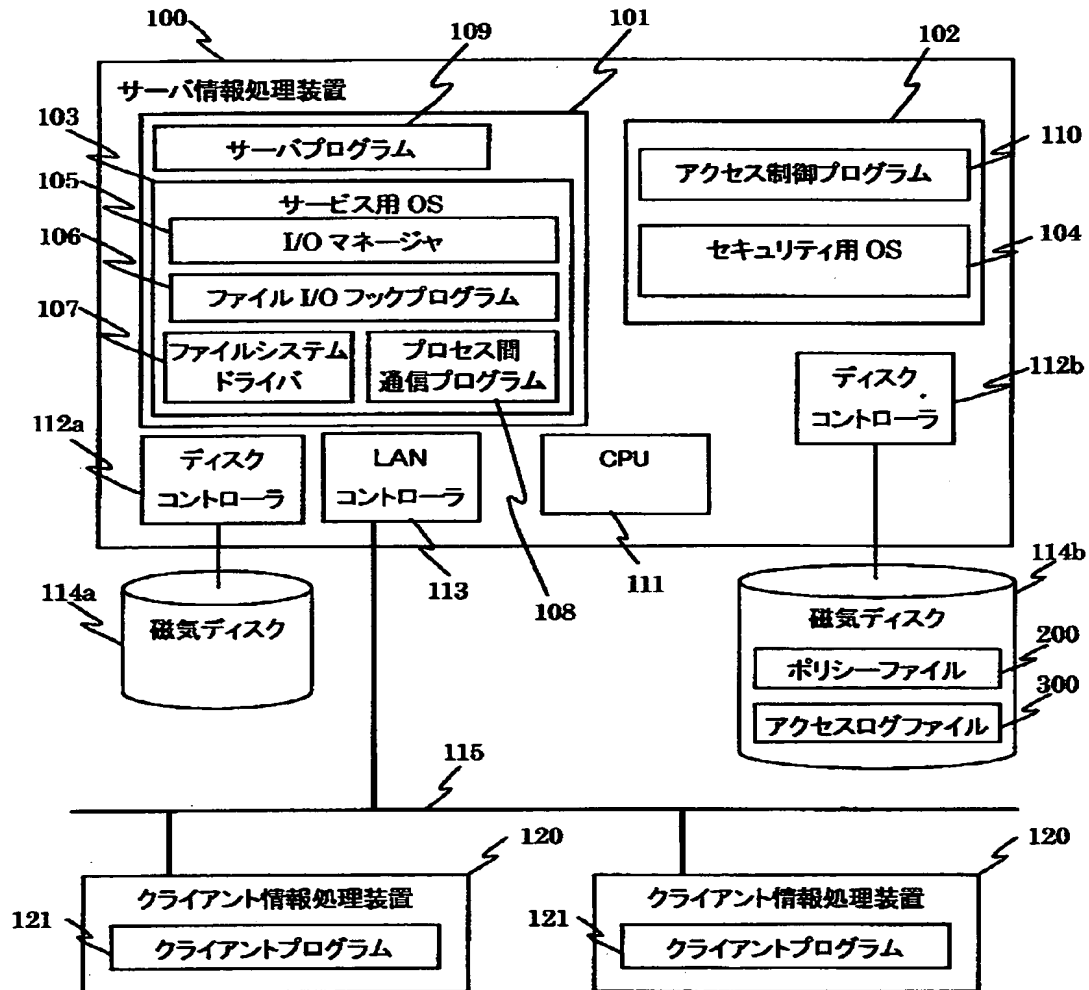
【符号の説明】

100…サーバ情報処理装置、101…サービス用OSが管理するメモリ、
102…セキュリティ用OSが管理するメモリ、103…サービス用OS、1
04…セキュリティ用OS、105…I/Oマネージャ、106…ファイル
I/Oフックプログラム、107…ファイルシステムドライバ、108…プロ
セス間通信プログラム、109…サーバプログラム、110…アクセス制御プ
ログラム、111…CPU、112…ディスクコントローラ、113…LA
Nコントローラ、114…磁気デバイス、115…LAN、120…クライ
アント情報処理装置、121…クライアントプログラム、200…ポリシーフ
ァイル、300…アクセスログファイル、501～506…通信経路
、600…I/Oパケット、610…リクエストパケット、620…レスポ
ンスパケット、1000…オープンファイルテーブル

【書類名】 図面

【図 1】

図 1



100: サーバ情報処理装置

120: クライアント情報処理装置

101: サービス用 OS が管理するメモリ

102: セキュリティ用 OS が管理するメモリ

【図 2】

図 2

200	201	202	203	204	205	206
	オブジェクト名	禁止されたアクセスのタイプ	エラーコード	例外サブジェクト	プログラムのハッシュ値	ユーザー名
210	D:\DOC\SECRET.TXT	Open	0018	c:\prog\wordedit.exe	0x22F0A412B73209CC	sec_admin
211		Delete	0021	c:\prog\fileman.exe	0x73209C2F0A1212B4C	sec_admin
	D:\SYS\CONFIG*	Write	0018	c:\prog\mantool.exe	0xB74122209FDE236C	sys_admin
		Delete	0021	c:\prog\mantool.exe	0xB74122209FDE236C	sys_admin
212	D:\LOG\LOG.TXT	Read	0016	c:\prog\audit.exe, c:\prog\evck.exe	0x2F21204B73C09A2C 0xF0A271249B8C202C	root, system
		Rename	0024	c:\prog\audit.exe	0x2F21204B73C09A2C	root

200: ポリシーファイル

【図 3】

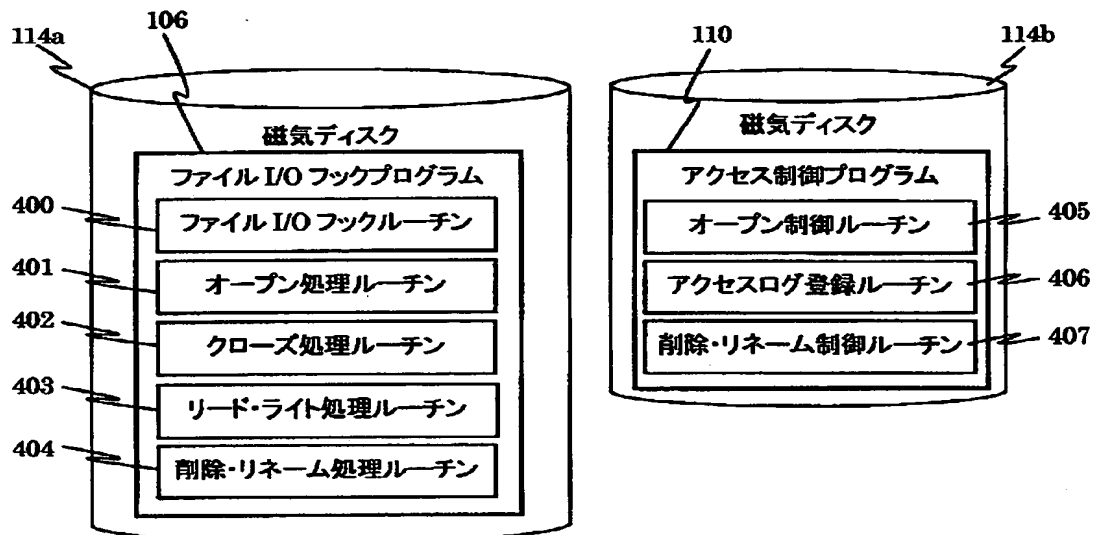
図 3

日時	オブジェクト名	アクセスタイプ	サブジェクト名	ユーザー名
1999.07.28 15:32:46	D:\DOC\SECRET.TXT	Write	c:\prog\wwwserv.exe	u_0023
1999.07.27 09:15:10	D:\DOC\SECRET.TXT	Delete	c:\prog\wwwserv.exe	u_0023
1999.07.25 16:02:55	D:\SYS\CONFIG\	Write	c:\prog\txtedit.exe	intruder
1999.07.25 14:44:28	D:\SYS\CONFIG\	Delete	c:\prog\fileman.exe	intruder
1999.07.25 14:42:59	D:\LOG\LOG.TXT	Read	c:\prog\txtedit.exe	user007
1999.07.16 10:29:31	D:\LOG\LOG.TXT	Delete	c:\prog\fileman.exe	user007

300: アクセスログファイル

【図 4】

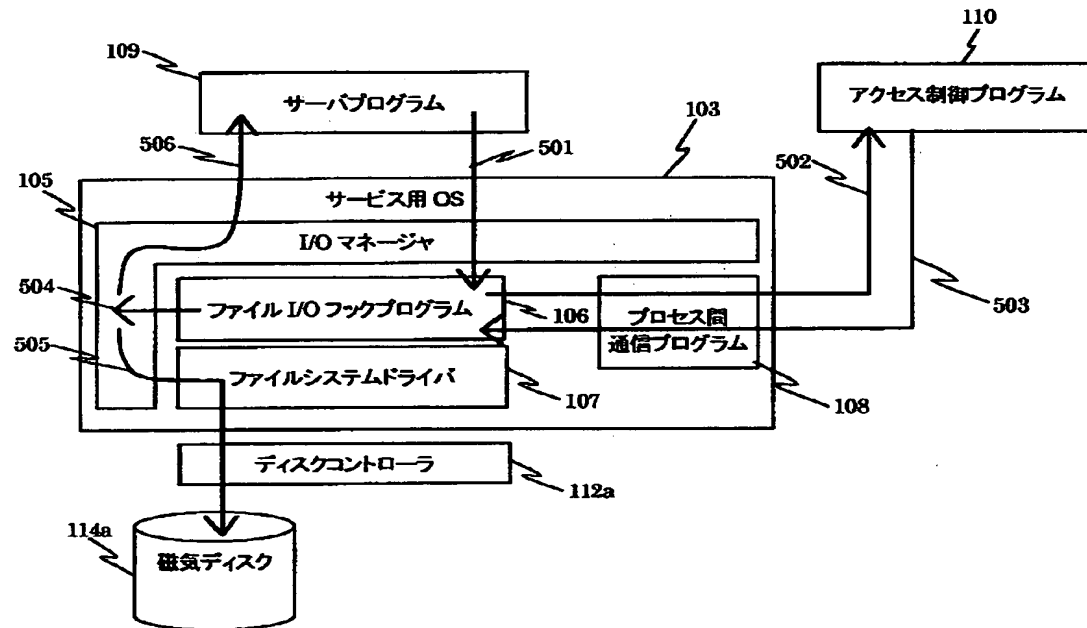
図 4



114a, 114b: 磁気ディスク

【図 5】

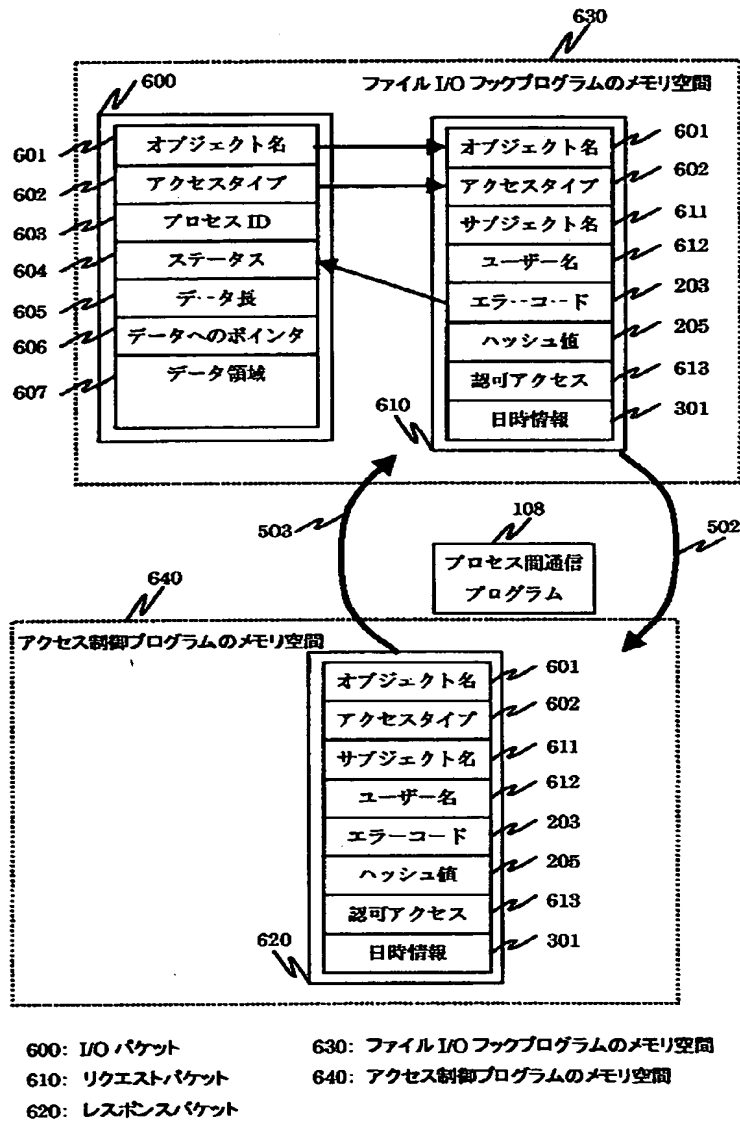
図 5



501~506: 通信経路

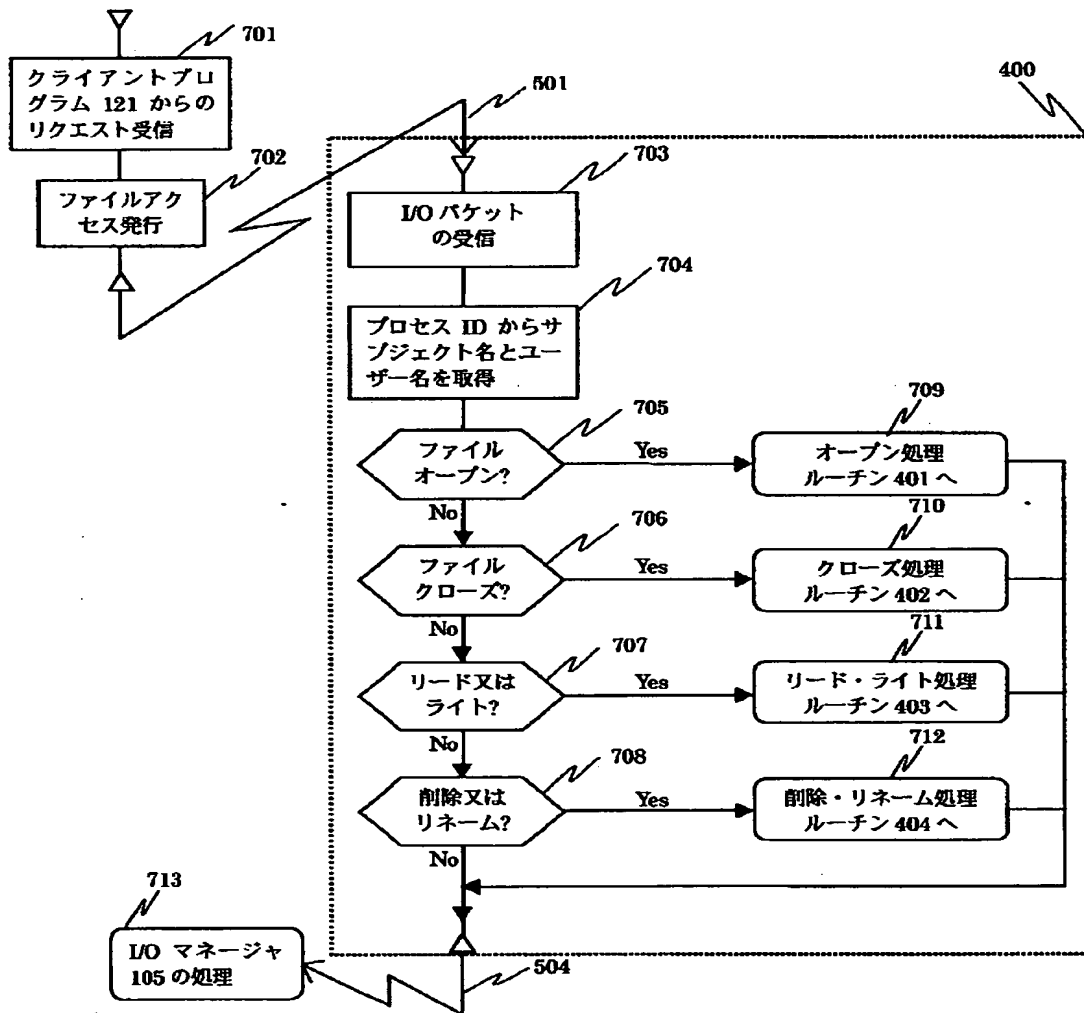
【図 6】

図 6

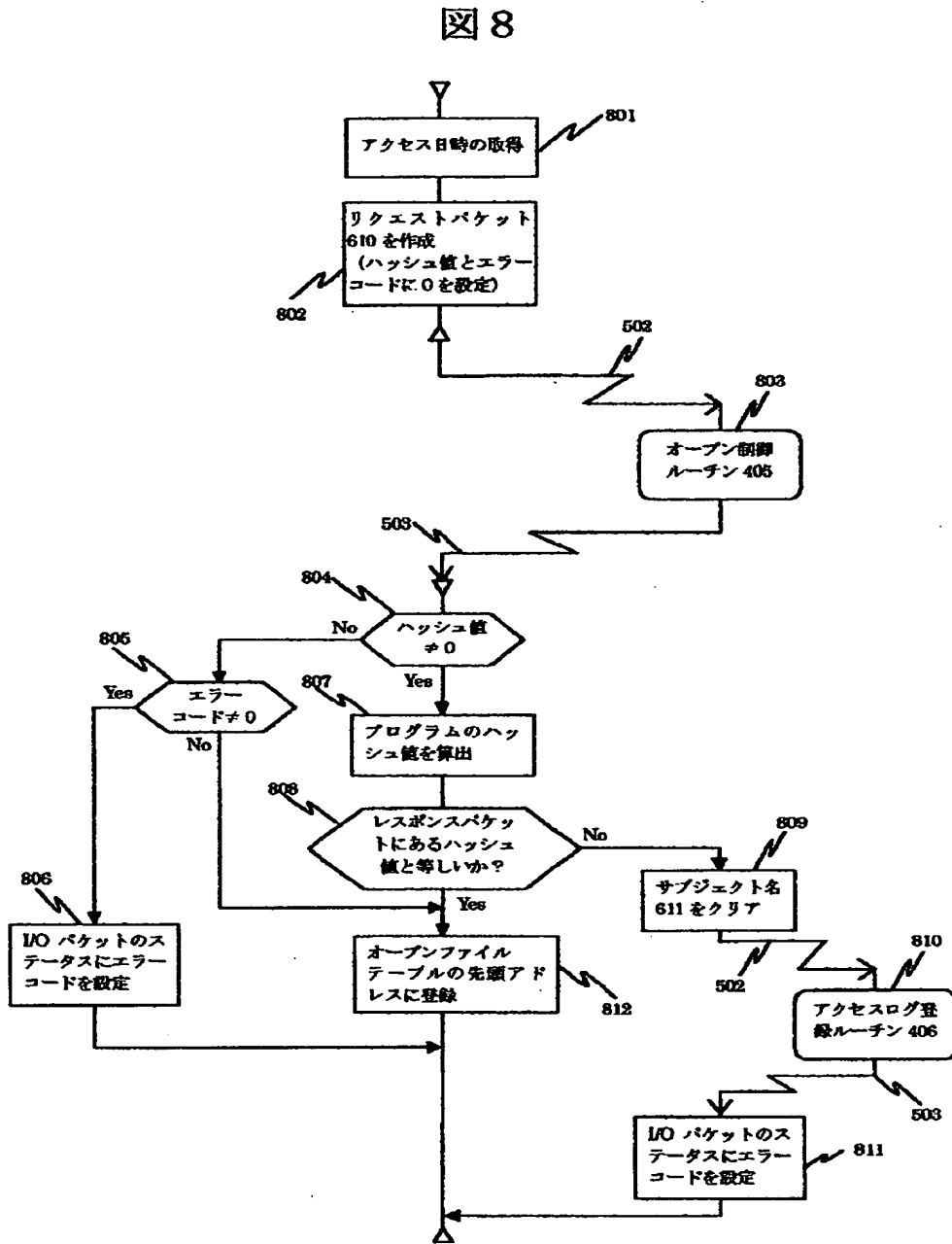


【図 7】

図 7

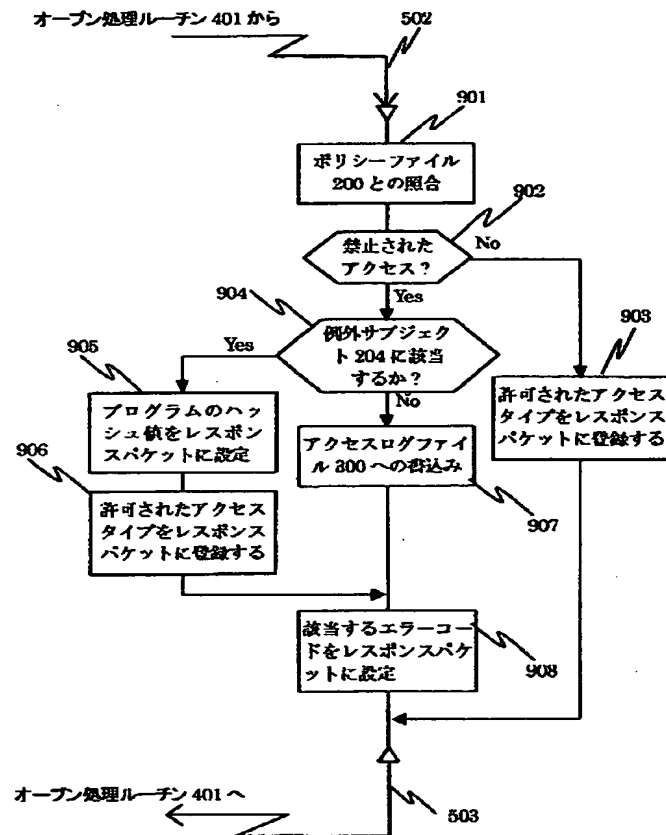


【図 8】



【図 9】

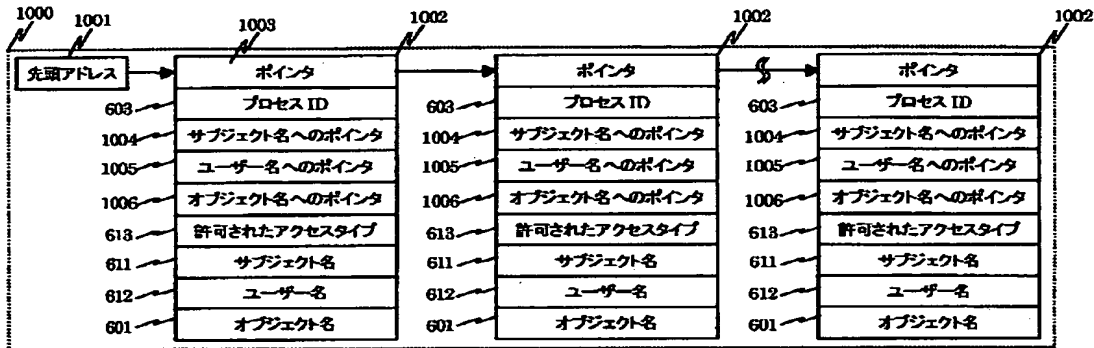
図 9



901～908: オープン制御ルーチン 405 の処理フロー

【図 10】

図 10



1000: オープンファイルテーブル
1002: オープンファイルの構造体データ

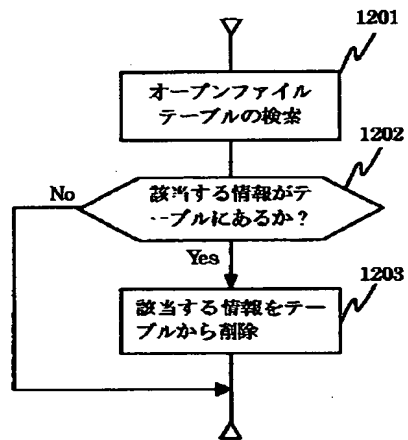
【図 11】

図 11

プロセス ID	サブジェクト名	ユーザー名	オブジェクト名	認可された アクセスタイプ
0022	c:\prog\wwwserv.exe	inet	D:\HOME\SALES.HTML	Read
0043	c:\prog\wordedit.exe	sec_admin	D:\DOC\SECRET.TXT	Read/Write
0067	c:\prog\viewer.exe	tarou	D:\DOC\IDEA.TXT	Read
0092	c:\prog\mantool.exe	sys_admin	D:\SYS\CONFIG\PORT	Read/Write
0113	c:\prog\vck.exe	system	D:\DOC\SHEET.DOC	Read
0218	c:\prog\audit.exe	root	D:\LOG\LOG.TXT	Read/Write

【図 1 2】

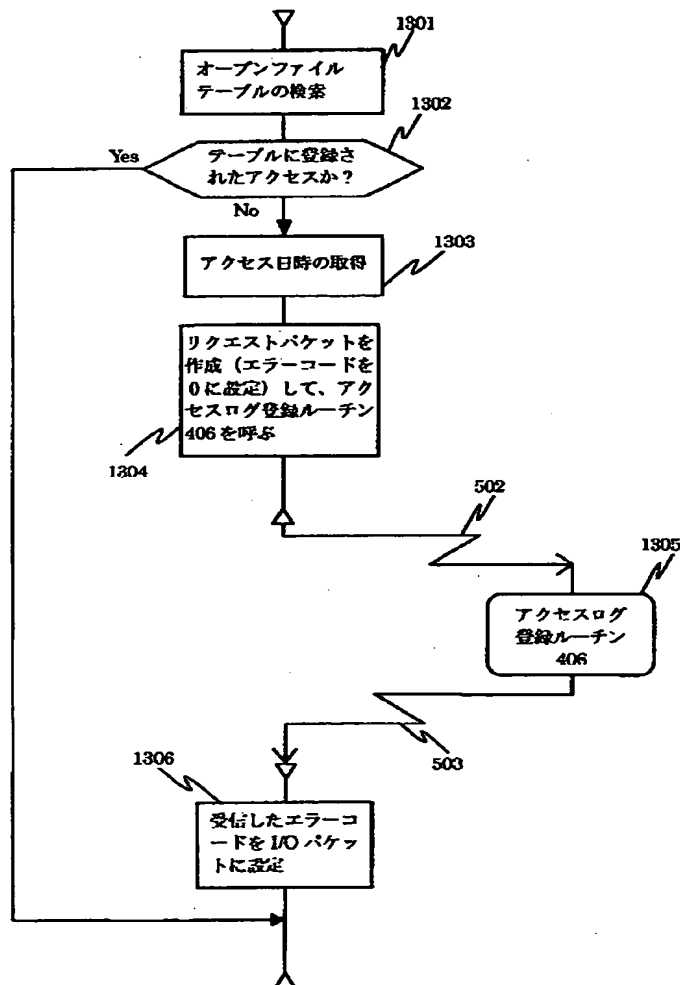
図 1 2



1201~1203: クローズ処理ルーチン 402 の処理フロー

【図 13】

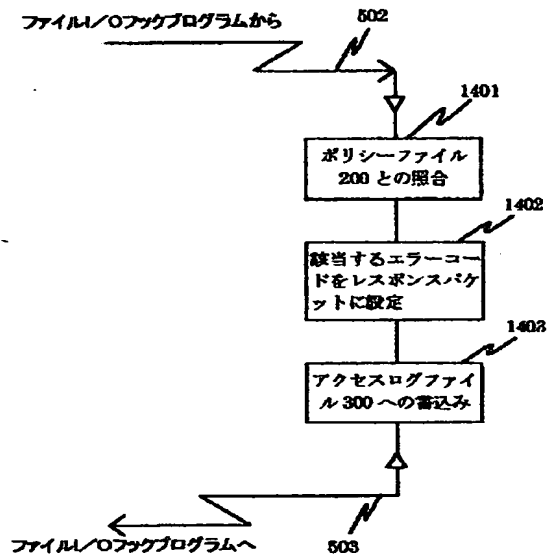
図 13



1301~1306: リード・ライト処理ルーチン 403 の処理フロー

【図 14】

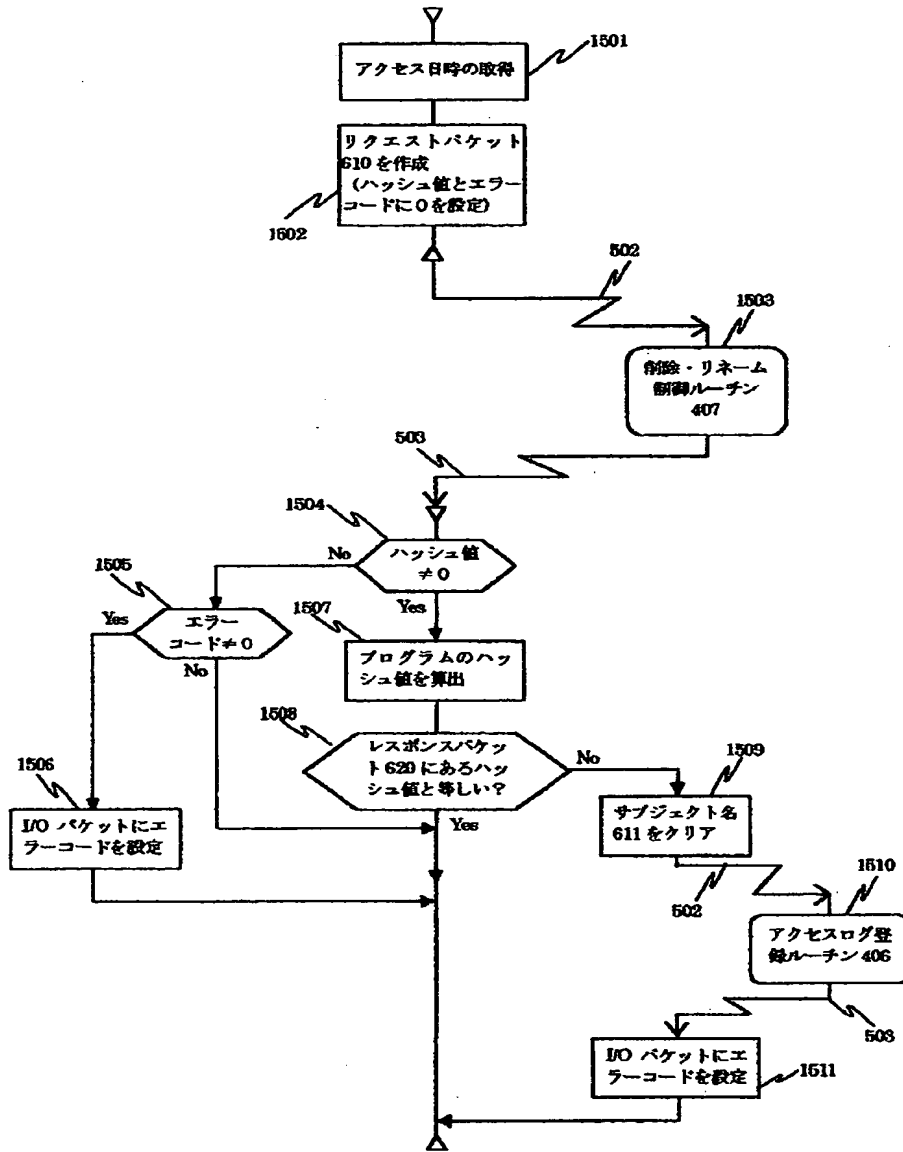
図 14



1401~1403: アクセスログ登録ルーチン 406 の処理フロー

【図15】

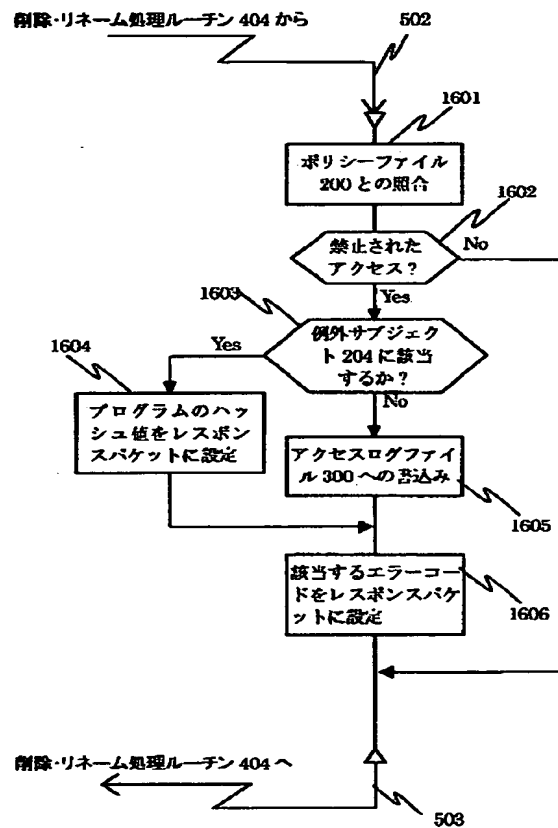
図15



1501～1509: 削除・リネーム処理ルーチン 404 の処理フロー

【図 16】

図 16



1601～1606: 削除・リネーム制御ルーチン 407 の処理シーケンス

【書類名】 要約書

【要約】

【課題】

ネットワークからの侵入者がいかなるユーザー権限を利用して不正なファイル読み出しや書き込みを試みても、該アクセスの抑止が可能なアクセス制御システム及びその方法を提供する。

【解決手段】

特定のファイルへのアクセスを、特定のユーザーが特定のプログラムを使用した場合に限り許可するといったポリシーを用いる。さらに、ポリシーをポリシーファイル200に登録し、ファイルI/Oフックプログラム106がフックしたアクセス情報を、プロセス間通信プログラム108を介してセキュリティ用OS104上のアクセス制御プログラム110に渡し、前記ポリシーに基づいたアクセス制御を行う。

【選択図】 図1

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日	1990年 8月31日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台4丁目6番地
氏 名	株式会社日立製作所